From: Mike O'Donnell
To: Microsoft ATR
Date: 1/23/02 7:10pm
Subject: Microsoft Settlement

[Text body exceeds maximum size of message body (8192 bytes). It has been converted to attachment.]

Renata B. Hesse
Antitrust Division
U.S. Department of Justice
601 D Street NW
Suite 1200
Washington, DC 20530-0001

Dear Ms. Hesse:

I would like to comment on the proposed Final Judgment in United States v. Microsoft, as provided in the Tunney Act.

I find that the proposed judgment is insufficient by a large margin to restore healthy competition in the computer operating systems and software application markets, so it is not in the public interest and should not be affirmed by the court.

The proposed Final Judgment attempts to remedy Microsoft's established illegal anticompetitive practices by prohibiting particular forms of conduct involving overly restrictive licensing terms, terms that vary in order to reward those who accept and punish those who contest a Microsoft monopoly, and terms that make switching to competing products more difficult or more costly. It also prohibits certain forms of retaliation against OEMs who support products competing with Microsoft's products. It also requires Microsoft to disclose APIs and communication protocols for its products under certain circumstances and for certain purposes.

It is inherently difficult, and perhaps impossible, to remedy Microsoft's particular forms of illegal anticompetitive behavior through conduct remedies. Both the underlying concepts in which conduct remedies are defined, and the particular anticompetitive techniques used by Microsoft change far too rapidly, and Microsoft itself has far too much influence on those changes, for them to serve in the foundation of effective conduct remedies.

The remedies in the proposed judgment refer to concepts of "API," "operating system," "middleware," "application," "platform software," "top-level window," "interface elements," "icons," "shortcuts," "menu entries." The definitions of these concepts are not robust and timeless. Compared to concepts in other branches of business and engineering they are relatively ephemeral, controversial, dependent on rapidly changing technological context, and subject to deliberate manipulation by Microsoft. For example, an "operating system" in the 1960s was a software system to organize the basic functionality of a computer, and it contained little or no user interface code. In the 1970s "operating systems" often contained substantial collections of utility applications and rudimentary interactive user interfaces

called "shells." In the 1980s, the X Window system was created as a form of what is now called "middleware" to provide a graphical interactive user interface, used widely in conjunction with Unix operating systems. Apple and Microsoft created similar graphical interactive user interfaces, but defined them to be parts of their operating systems, rather than additional middleware. In the near future, distributed and network computing are likely to make it quite difficult to determine the boundaries of a single operating system. In the past, Microsoft appears to have deliberately manipulated the boundaries of such conceptual categories to create and preserve a monopoly position, and I expect it to continue such practices in the future. The proposed judgment provides definitions that narrow these already problematic concepts even further, making them even more vulnerable to deterioration due to technological change and to manipulation by Microsoft.

Furthermore, the particular conduct requirements in the proposed judgment are far too narrow. Every one of the requirements is weak in some way. For example, consider the requirement to "disclose to ISVs, IHVs, IAPs, ICPs, and OEMs, for the sole purpose of interoperating with a Windows Operating System Product, ... the APIs and related Documentation that are used by Microsoft Middleware to interoperate with a Windows Operating System Product." Microsoft and other software vendors like to treat their Applications Product Interfaces (API) as intellectual property. But in good engineering practice these are key parts of the warrantable specifications of a product. This holds in particular for operating systems and middleware, which by their nature are especially intended for, suitable for, and often useless without interaction with other software products. APIs define the quality of that interaction, but they do not provide it. The implementation of an API in program code (which is naturally protected by trade secret, copyright, and patent law) provides the quality of interaction defined by an API. Without access to the complete API, the licensor of an operating system cannot employ the system freely in the way that good software engineering practice suggests. With complete public access to an API, a software company may still protect its implementation of the API, which contains the real value that it has created. Keeping an API secret does not correspond to keeping the inner workings of a product secret. Rather, it corresponds to keeping the precise function accomplished by that product secret.

So the public interest calls for the widest possible dissemination of API documentation. But the proposed judgment explicitly calls for disclosure of APIs "for the sole purpose of interoperating with a Windows Operating System Product," and only the "APIS and related Documentation that are used by Microsoft Middleware to interoperate with a Windows Operating System Product." This excludes the use of information about the API to provide competitive platforms for running Windows-compatible software. Keep in mind that Windows-compatible

software does not necessarily come from Microsoft. Microsoft benefits from the value added to its operating system products by a large number of less powerful software houses that create Windows-compatible software. By holding the Windows operating system API secret, Microsoft in effect keeps crucial information about other companies' software applications secret, denying those applications the value added by competing operating systems on which they may run.

Compare the Windows market (and the preceding DOS market) to the Unix/Linux/Posix market. Microsoft uses secret and changeable APIs to effectively eliminate competition to provide alternative operating systems running Windows applications. A competing operating system must use different APIs, and therefore cannot support all of the same applications. By contrast, the Posix standard is a completely public API for Unix/Linux. Various companies, such as Sun Microsystems, compete to provide different implementations of the Posix API. Consumers may run Unix/Linux applications on any of these operating systems.

Similarly, in the hardware market for processors, the specification of the x86 instruction set architecture (the hardware analog to a software API), is public. As a result, AMD competes with Intel to implement that architecture, with immense benefit to the public interest. Similar publication of standards in the overall functionality of personal computers led to the immensely beneficial competition among makers of IBM-compatible PCs. The failure to disclose Windows operating system APIs destroys the possibility of similarly beneficial competition among vendors of operating systems.

Very similar considerations to those raised above for APIs apply to communication protocols (for which the proposed judgment provides limited disclosure) and to file formats (not covered in the proposed judgment). Note that Adobe made full public disclosure of its PostScript and PDF formats, compared to Microsoft's secrecy regarding Word formats, and that this disclosure served the public interest immensely by promoting the wide availability of PostScript and PDF printers and viewers.

There are many other detailed shortcomings of the proposed Final Judgment, including the remaining conduct restrictions and the enforcement methods. I expect that other correspondents will treat some of them.

Sincerely yours,

Michael J. O'Donnell

Professor in Computer Science and the Physical Sciences Collegiate Division
The University of Chicago

Senior Fellow in the Computation Institute of
The University of Chicago and Argonne National Laboratory